

Práctico 2: Git y GitHub

- 1) GitHub es una plataforma que permite el alojamiento de proyectos que utilizan git, un sistema de control de versiones.
- 2) Primer hay que crear un usuario e iniciar sesión con el. Luego en tu perfil esta la sección “tus repositorios” y ahí se mismo se crea uno donde se puede configurar nombre, si quiero que sea publico o privado y demás opciones.
- 3) Para crear una rama hay que utilizar “git Branch + nombre de la rama” que quiero utilizar’
- 4) Para cambiar de ramas se utiliza “git checkout + el nombre de la rama” a la cual quiero ir.
- 5) Primero me paro en la rama que quiero fusionar y luego uso “git merge rama-origen + rama-destino”
- 6) Para crear un commit primero hay que agregar al Stage los archivos cuales quiero versionar con “git add .” luego “git commit -m + nombre del commit”
- 7) Una vez realizado el commit en el repo local. Hay que conectar el repositorio local con el remoto y por ultimo hacer “git push -u origin master”
- 8) Es lo mismo que el repositorio local, pero almacenado en la nube lo cual permite el acceso desde cualquier terminal siempre y cuando se tenga las credenciales.
- 9) Con el código “git remote add origin + url del repo remoto” se conecta el repo remoto con el local.
- 10) Una vez guardados los cambios en mi repo local con “commit” utilizo la línea de código “git push -u origin master” para enviar los cambios al repo remoto.
- 11) Para traer los cambios de un repo remoto al local utilizo la línea “git pull origin master”
- 12) Es una copia completa de un repositorio que se encuentra en una cuenta de usuario diferente en plataformas como GitHub.
- 13) Una vez iniciada sesión en la cuenta de github, tengo que ir al repositorio que deseo hacer el fork. Hacer click en el botón “Fork” y luego seleccionar la cuenta de destino.
- 14) Con mi repositorio “forkeado”, realizado un “clone” al repo local, hecho cambios y luego agregarlos “add” y “commit” y por ultimo con “push” subirlos al repo remoto. Voy a mi repo remoto a la pestaña “pull request” y luego en “new”.

- 15) Para aceptar una solicitud de “request” hay que ir al repo donde se a realizado el “pull request” y ver la lista de pendientes para revisar y aceptar.
- 16) Una etiqueta es un marcador que se utiliza para marcar un punto especifico en la historia del repositorio. Son muy útiles por ejemplo para identificar versiones de software ya que permite moverse de la v1.0 a v2.0.
- 17) Para crear una etiqueta se utiliza “git tag + nombre”
- 18) A diferencia del git push que envia los commits al repo remoto, para enviar los tags hay que hacerlo de forma explicita con la línea de código “git push origin + nombre etiqueta”
- 19) El historial de git es un registro detallado de todos los cambios realizados en un repositorio de Git a lo largo del tiempo. Este historial incluye todos los commits, las ramas, las etiquetas y cualquier otra acción que haya sido registrada en el repositorio.
- 20) Puedo acceder al mismo con “git log”
- 21) Existen distintas maneras de buscar en el historial de git. Ellas pueden ser a través de los tags, commits, palabra especifica o hasta un rango de fechas.
- 22) Para borrar el historial de git hay que crear una nueva rama “origen” con el estado actual del repo. Luego eliminar el historial y hacer un commit desde el inicio.
- 23) A diferencia del publico donde cualquiera puede ver el código, este es de acceso limitado previa autorización. Teniendo administradores que gestionan quien puede acceder o no.
- 24) Dentro de las configuraciones básicas al momento de crear un repositorio remoto en github no preguntara si queremos que sea publico o privado.
- 25) Para invitar a un usuario (Tiene que tener una cuenta activa) hay que ir al repo remoto y en la sección colaboradores buscar el nombre de el usuario y apretar “agregar o invitar”
- 26) Como mencione en el punto “23” el repositorio publico permite que cualquier pueda ver el código y las colaboraciones.
- 27) Al momento del crear el repo remoto poner la opción “Publico”
- 28) Para compartirlo simplemente copiar la url del repositorio.

Actividad 2)

<https://github.com/Damian674/ejercicio2>

Actividad 3)

<https://github.com/Damian674/conflict-exercise.git>